#include <iostream>

#include <string>

using namespace std;

// 1. Add an accept(Visitor) method to the "element" hierarchy

class Element

{

public:

virtual void accept(class Visitor &v) = 0;

};

class Company: public Element

{

public:

void accept(Visitor &v);

int profit[4];

void init()

{

profit[0] = 2;

profit[1] = -1;

profit[2] = 10;

profit[3] = 3;

}

int report()

{

return 2+(-1)+5+3;

}

};

// 2. Create a "visitor" base class w/ a visit() method for every "element" type

class Visitor

{

public:

virtual void visit(Company \*e) = 0;

};

void Company::accept(class Visitor &v)

{

v.visit(this);

}

// 3. Create a "visitor" derived class for each "operation" to do on "elements"

class TaxVisitor: public Visitor

{

/\*virtual\*/void visit(Company \*e)

{

int k= e->report();

cout <<k << "?\n";

cout << "What is real value of Quarter 3\n";

}

};

class PRVisitor: public Visitor

{

/\*virtual\*/void visit(Company \*e)

{

int k= e->report();

cout <<k << "!!!\n";

e->report();

cout << "This company realy earns " << 2+0+5+3;

}

};

int main()

{

Company c;

c.init();

//cout << c.profit[2];

TaxVisitor t;

PRVisitor p;

c.accept(t);

c.accept(p);

};

**Exercise 1:**

A building inspector may visit many different buildings. At each building, the inspector wants to know the progress, the amount paid and the remaining amount. Write a simple program that simulates the inspection process ***I*** to buildings ***A*** and ***B***.

**Exercise 2:**

After the inspector examines the buildings, he must report the results to two departments, the project management department and the finance department, on the results obtained. Write a program to simulate this.